

Cours n°12 du 29 novembre 2012

3MCI n1 2012-2013

Cycle de vie des ViewController
UIImageView
UIWebView

Cycle de vie des ViewController

Cycle de vie des ViewController

► **Les ViewController ont un cycle de vie:**

- ils naissent (sont créés)
- ils existent pendant une certaine durée
- ils meurent (disparaissent, sont vidés de la mémoire)

► **Les phases de ce cycle sont prévisibles:**

- un certain nombre de messages leur sont envoyés à ces différentes étapes

Redéfinition de méthode

En anglais, *overriding method*

- ▶ **Héritage: une sous-classe hérite des méthodes de sa super-classe**
 - parfois, elle les réutilise telles quelles
 - souvent elle a besoin de les modifier. Dans ce cas, deux solutions:
 - créer une nouvelle méthode avec un nouveau nom
 - **redéfinir** la méthode existante dans la super-classe
- ▶ **Deux règles pour cette redéfinition (ou surcharge):**
 - la méthode redéfinie doit prendre le même nombre et type d'arguments que la méthode de la super-classe
 - elle doit avoir le même type de retour que la méthode de la super-classe

5

Cycle de vie des ViewController

- ▶ **Les phases de ce cycle sont prévisibles:**
 - un certain nombre de messages leur sont envoyés à ces différentes étapes

— [**On va très souvent redéfinir un certain nombre de ces méthodes**

- ➔ pour réaliser certaines tâches

6

Cycle de vie des ViewController

à la création

► Création

- est faite par un Segue ou directement par le storyboard
 - méthode `instantiateViewControllerWithIdentifier:`
- n'est quasiment jamais redéfinie

7

Cycle de vie des ViewController

quand il va venir à l'écran...

– (void)viewDidLoad

- est appelé après l'instanciation du ViewController et lorsque ses outlets sont définis
- excellent endroit pour placer du code lié à la mise en place du Controller
- attention: la géométrie de la View (ses bounds) n'est pas encore définie!
 - si on veut lancer quelque chose en rapport avec la géométrie, utiliser la méthode suivante...

8

Cycle de vie des ViewController

quand il va venir à l'écran...

– (void)viewWillAppear:(BOOL)animated

- est appelé juste avant que la View apparaisse à l'écran
- les bounds ont été définis selon les frames fixés par la superView
- endroit idéal pour lancer quelque chose de coûteux (en processeur, p. ex.)
 - si coûteux en temps (charger du contenu du réseau), le faire plus tôt pour éviter un blocage
- on commence souvent par appeler la méthode de la super-classe:

```
[super viewWillAppear:animated];
```

9

Cycle de vie des ViewController

quand il va disparaître de l'écran...

– (void)viewWillDisappear:(BOOL)animated

- est appelé juste avant que la View disparaisse de l'écran
- c'est un bon endroit pour sauvegarder des informations, l'état de l'application

```
{  
[super viewWillDisappear:animated];  
    //On appelle d'abord la version de la super-classe  
[self rememberScrollPosition];  
    // pour se rappeler de la position du scroll dans la View  
[self saveDataToPermanentStore];  
    // sauvegarder des données  
}
```

10

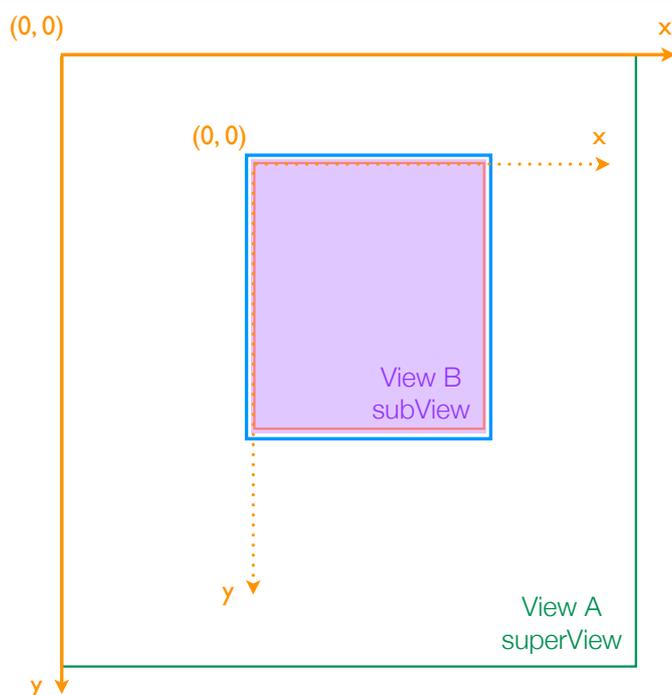
Cycle de vie des ViewController

bien plus encore...

▸ Il existe encore de nombreuses autres méthodes

- (void)viewDidAppear:(BOOL)animated;
- (void)viewDidDisappear:(BOOL)animated;
- (void)willLayoutSubviews;
- (void)viewDidLayoutSubviews;
- (void)willRotateToInterfaceOrientation:(UIInterfaceOrientation)anOrientation
duration:(NSTimeInterval)seconds;
- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)anOrientation;
- (void)viewDidUnload
- ...

11



▸ **frame** de la **View B**:

- un rectangle (CGRect, pt d'origine et taille) qui contient entièrement la **View B** dans le système de coordonnées de la **View A**
- ((50, 25), (75, 100))

▸ **bounds** de la **View B**:

- CGRect, pt d'origine et taille de la View dans son propre système de coordonnées
- ((0, 0), (75, 100))

Qui est responsable de fixer le frame d'une View?

L'objet qui place la View dans la hiérarchie des Views

12

UIImageView

13

UIImageView

▸ sous-classe de UIView qui sert à afficher une UIImage

- UIImage est un objet que l'on peut créer à partir de notre dossier «ressources»
- `UIImage *image = [UIImage imageNamed:@"monImage.jpg"];`

▸ comment fixer la UIImage d'un UIImageView?

```
- (id)initWithImage:(UIImage *)image;  
  
// va faire correspondre la taille du frame de la View à la taille de l'image  
  
@property (nonatomic, strong) UIImage *image;  
  
// la taille de la frame ne sera pas ajustée à la taille de l'image
```

▸ UIView @property.contentMode

- détermine où l'image apparaît dans la View et si elle est mise à l'échelle ou non

14

UIWebView

UIWebView

- ▶ **un navigateur internet complet intégré dans une UIView**
- ▶ **basé sur webkit (moteur de rendu HTML opensource utilisé dans Safari et Chrome, notamment)**
- ▶ **supporte Javascript, limité à 5 secondes et 10 Mo de mémoire**

UIWebView

▸ **pour charger du HTML dans la View, on envoie le message:**

– (void)loadRequest:(NSURLRequest *)request;

▸ **NSURLRequest: classe permettant d'effectuer une requête en fonction d'un URL**

- pour obtenir un NSURLRequest, on peut utiliser une méthode de classe:
+ (NSURLRequest *)requestWithURL:(NSURL *)url;
- indépendante du protocole ou du type d'URL

17

UIWebView

▸ **NSURL est une classe dont les objets sont quasiment comme une NSString, sauf qu'on les force à être «correctement formés»**

- exemple: http://... , ou file://... ou encore ftp://...
- c'est la manière recommandée pour spécifier le nom d'un fichier en iOS
- pour obtenir un NSURL, on utilise une méthode de classe:
+ (NSURL *)urlWithString:(NSString *)urlString;

▸ **en résumé, voici comment on demande à une UIWebView de charger une certaine adresse URL:**

```
[self.webView loadRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:@"http://www.monsiteweb.ch"]]];
```

18